NPS-MA-97-002

NAVAL POSTGRADUATE SCHOOL
Monterey, California

A Partial Ordering of the Chordal Graphs

by

Craig W. Rasmussen
LCDR Thomas Carroll, USN

February 1997

Approved for public release; distribution is unlimited.

Prepared for: Naval Postgraduate School
Monterey, CA 93943-5000

19970314 052

DTIC QUALITY INSPECTED 1

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943

Rear Admiral M. J. Evans
Superintendent

Richard Elster
Provost

This report was prepared in conjunction with research conducted for the Naval Postgraduate School and funded by the Naval Postgraduate School.

Reproduction of all or part of this report is authorized.

This report was prepared by:

Craig Rasmussen
Assistant Professor of Mathematics

Reviewed by:

WALTER M. WOODS
Chairman

Released by:

DAVID W. NETZER
Associate Provost and Dean of Research

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
7 February 1997

3. REPORT TYPE AND DATES COVERED
Technical Report January 96 - March 96

4. TITLE AND SUBTITLE
A Partial Ordering of the Chordal Graphs

5. FUNDING NUMBERS
N/A

6. AUTHOR(S)
Craig Rasmussen and LCDR Thomas Carroll, USN

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER
NPS-MA-97-002

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The chordal graphs have been well-studied because of their desirable algorithmic characteristics. Many problems that are intractable in the general case are solvable by fast algorithms in the chordal case. We show that the chordal graphs are partially ordered under edge set inclusion, and describe algorithms for bidirectional traversal of maximal chains in the corresponding cover graph. We also describe the embedding of several subclasses of the chordal graphs as subposets of the chordal poset, and suggest application of these order relations to the design of improved heuristics for obtaining approximate solutions to problems on arbitrary graphs.

14. SUBJECT TERMS
chordal graphs, graph completions, elimination orderings ordered sets

15. NUMBER OF PAGES
12

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std 239-18
298-102

A PARTIAL ORDERING OF THE CHORDAL GRAPHS

LCDR THOMAS CARROLL, USN*
CRAIG W. RASMUSSEN†
DEPARTMENT OF MATHEMATICS
NAVAL POSTGRADUATE SCHOOL

Abstract. The chordal graphs have been well-studied because of their desirable algorithmic characteristics. Many problems that are intractable in the general case are solvable by fast algorithms in the chordal case. We show that the chordal graphs are partially ordered under edge set inclusion, and describe algorithms for bidirectional traversal of maximal chains in the corresponding cover graph. We also describe the embedding of several subclasses of the chordal graphs as subposets of the chordal poset, and suggest application of these order relations to the design of improved heuristics for obtaining approximate solutions to problems on arbitrary graphs.
Keywords: Chordal graphs, graph completions, elimination orderings, ordered sets.

1. Preliminaries. We consider only simple graphs, allowing neither loops nor multiple edges. If $G = (V, E)$ is a graph, and if $W \subseteq V$, then we denote by $\langle W \rangle$ the subgraph of G induced by W. Given a graph $G = (V, E)$, we denote by $\overline{G}$ the *complement* of G, i.e., $\overline{G} = (V, \overline{E})$, where $\overline{E} = \{xy | x, y \in V, xy \notin E\}$. We use $X + Y$ to denote the union of disjoint sets X, Y. If $G = (V, E)$, $F \subseteq E(\overline{G})$, and $f \in F$, then we denote by $G + f$ the graph $G' = (V, E + \{f\})$. Similarly, if $e \in E$, we denote by $G - e$ the graph $G' = (V, E \backslash \{e\})$. A graph $G = (V, E)$ is *chordal* if G contains no induced k-cycle for $k \geq 4$. The *neighborhood* $N(v)$ of a vertex $v \in V$ is defined by $N(v) = \{x \in V | vx \in E\}$; the *closed neighborhood* of v is given by $N[v] = N(v) + \{v\}$. A vertex v in G is *simplicial* if $\langle N(v) \rangle$ is a *clique*, i.e., a complete subgraph. If $V' \subseteq V$, $V' \neq \emptyset$, and $\langle V' \rangle = (V', \emptyset)$, then V' is an *independent set*. We denote by $\omega(G)$ the order of a largest clique in G, and by $\chi(G)$ the chromatic number of G. For terminology not defined here, see Bondy [1] or West [15].

Given a family $\mathcal{F} = \{F_1, F_2, \ldots, F_k\}$ of subsets of some universal set, we may construct the *intersection graph* G of $\mathcal{F}$ by letting the vertices of G be the elements of $\mathcal{F}$ and including the edge F_iF_j if and only if $i \neq j$ and $F_i \cap F_j \neq \emptyset$. An *interval graph* is a graph G that can be represented as the intersection graph of a family of intervals on the real line. If this can be accomplished using intervals of constant length, G is said to be *unit interval*. For details on constructing (unit) interval representations of (unit) interval graphs, see Roberts [12, 13].

It is a well-known fact, first reported by Dirac [5], that every chordal graph possesses a simplicial vertex. The existence of such a vertex, and the fact that chordality is hereditary, make possible an efficient recognition algorithm for chordal graphs due to

* MUCH OF THE WORK DESCRIBED HERE APPEARS IN LCDR CARROLL'S M.S. THESIS [2].

† CODE MA/RA, DEPARTMENT OF MATHEMETICS, NAVAL POSTGRADUATE SCHOOL, MONTEREY, CA 93943

Fulkerson and Gross [7]. They show that every chordal graph G has a *perfect elimination ordering*, a labeling of the vertices as $v_1, v_2, \ldots, v_n$ such that for each $1 \leq i \leq n$, v_i is simplicial in $\langle v_i, v_{i+1}, \ldots, v_n \rangle$; moreover, they show that only chordal graphs possess such orderings.

An *n-sun*, $n \geq 3$, is a chordal graph G on $2n$ vertices that can be labeled as $x_1, x_2, \ldots, x_n$, $y_1, y_2, \ldots, y_n$ such that $x_1, x_2, \ldots, x_n, x_1$ is a cycle (not necessarily induced), $y_1, y_2, \ldots, y_n$ is an independent set, and $N(y_i) = \{x_i, x_{i+1}\}$ for each $i = 1, 2, \ldots, n$. The addition in subscripts is understood to be cyclic. A *strongly chordal graph* is most simply described as a chordal graph that contains no induced n-sun. An efficient recognition algorithm and a number of alternative characterizations can be found in Farber [6].

A *split graph* is a graph $G = (V, E)$ with the property that V can be partitioned as $V = K + I$ in such a way that $\langle K \rangle$ is a clique and $\langle I \rangle$ is an independent set. It follows immediately from the definition that G is a split graph if and only if $\overline{G}$ is a split graph. It is easy to see that split graphs are chordal, since no cycle on more than three vertices allows such a partition. A subclass of the split graphs is the class of *threshold graphs*. A number of characterizations exist; that which we use relies upon a partition of the vertices of $G = (V, E)$ in terms of degree. The *degree partition* of V is defined in the following way. If the positive degrees of the vertices of G are $d_1, d_2, \ldots, d_m$, we let D_i contain all vertices of degree d_i for each $1 \leq i \leq m$, and place any isolated vertices in D_0. Then $V = D_0 + D_1 + \cdots + D_m$ is the degree partition of V. By a result of Chvátal and Hammer [3], G is threshold if and only if for all $x \in D_i$, $y \in D_j$, $xy \in E$ if and only if $i + j > m$. For other characterizations and applications of threshold graphs, as well as additional references, see Golumbic [8].

A graph G is said to be *perfect* if $\chi(G') = \omega(G')$ for all induced subgraphs G' of G. Many well-known classes of graphs fall into the class of perfect graphs, including the chordal graphs and the bipartite graphs. Certain classes of perfect graphs arise in applications, and many classes of perfect graphs have desirable algorithmic properties. For a compilation of much that is known about perfect graphs, see Golumbic [8]. Each class of graphs discussed here is a perfect class.

Let $G = (V, E)$ be a graph of order n and size q, and suppose that G has some property P. We say that G is *P-completable* if the edges of $\overline{E}$ can be added serially in such a way that each supergraph in the resulting sequence has property P; the associated ordering $e_1, e_2, \ldots, e_k$, where $k = \binom{n}{2} - q$, is a *P-completion sequence* or, when no ambiguity is likely to arise, simply a *completion sequence*.

If all graphs with property P are P-completable, we say that the class Π of graphs with property P is a *completion class*. Some properties are such that *any* ordering of the edges missing from a representative G constitutes a completion sequence. For example, the class of all connected graphs on p vertices for fixed p is a trivial completion class. We are interested in nontrivial cases. In general, we shall refer to such classes as *conditional* completion classes; since only conditional completion classes will be considered here, we shall omit the adjective.

The first appearance in the literature of the idea of a conditional graph completion

appeared in the work of Grone, Johnson, et al. [9], who showed that the chordal graphs constitute a completion class. The proof was constructive and relied on earlier work by Rose, Tarjan, and Lueker [14]. In a previous paper [11], one of the present authors shows that a number of additional classes of perfect graphs, most of them chordal, are also completion classes. In [10] we add the strongly chordal graphs to the list of completion classes, and describe algorithms that use elimination orderings to construct completion sequences. Here we extend those results. In particular, we describe polynomial-time algorithms with which *annihilation sequences* can be found for a number of these classes. While the reversal of a completion sequence is an annihilation sequence, one must consider edges in a different order to construct such a sequence directly.

As a consequence of these results, we show that the chordal graphs of fixed order, and many of the subclasses thereof, are in some sense broadly distributed among the graphs of that order. Moreover, the chordal graphs and the subclasses studied are shown to be partially ordered under edge set inclusion.

2. Annihilation Sequence Construction. In [11], we show that the interval, unit interval, threshold, and split graphs, among others, are completion classes. While the proofs of those results are constructive, it is existence rather than efficient construction of such sequences that is established. The naive algorithms suggested in [11] so not, in general, allow efficient implementation. In [10], we refine those results to show that completion sequences for these graphs can be quite efficiently obtained. We show also that similar results can be obtained for the strongly chordal graphs, obtaining as a corollary that these also constitute a completion class.

The algorithms for generating completion sequences follow the lead of Grone et al. [9] in that they exploit the existence of characteristic elimination orderings for each subclass. The fundamental algorithm (see Algorithm A) accepts as input a graph $G = (V, E)$; by choosing the appropriate ordering of the vertices, the sequence of edges produced is a completion sequence of some prescribed type.

This algorithm suffices for constructing completion sequences for interval, split, strongly chordal, and threshold graphs. For unit interval graphs, the algorithm requires some modification.

Note that the algorithm can be implemented by a simple nested loop in which each of the $\binom{n}{2}$ possible edges is considered exactly once, the order in which they are considered completely determined by the labeling. The algorithm is consequently quite efficient. While it is clear from earlier work [10] that alterations to the chordal and strongly chordal completion algorithms will result in annihilation algorithms, the details have not been presented. In this report, we provide those details, and present algorithms for constructing annihilation algorithms that preserve membership in the remaining classes under consideration.

We use two variations on the theme illustrated by Algorithm A to construct these annihilation sequences. Each accepts as input a nonempty graph $G = (V, E)$ and each iteratively deletes edges until the remaining graph is empty. Each algorithm chooses from among those vertices with positive degree the vertex v_i with the least index. The edges incident to v_i are then sequentially deleted until v_i is isolated. The first

Algorithm A

Input: Graph $G = (V, E)$ of order n, size q, with vertex labeling $\alpha = v_1, \ldots, v_n$.

Output: K_n.

```
BEGIN
   G_0 := G;
   E_0 := E;
   s := (n choose 2) − q;
   FOR i := 1 TO s DO
      BEGIN
         k_i := max{k | deg_{i−1} v_k < n − 1};
         j_i := max{j | v_j v_{k_i} ∉ E_{i−1}};
         e_i := v_{k_i} v_{j_i};
         E_i := E_{i−1} + e_i;
         G_i := (V, E_i);
      ENDFOR
END.
```

FIG. 1. *Generic completion algorithm*

variation, which we here call Algorithm B, chooses at each pass the edge v_iv_j satisfying $j = \min\{k|v_iv_k \in E\}$, while the second variation (Algorithm C) chooses the edge v_iv_j satisfying $j = \max\{k|v_iv_k \in E\}$.

We show in the following sections that one or the other of these algorithms suffices to construct annihilation sequences for the chordal, strongly chordal, interval, unit interval, split, and threshold graphs.

2.1. Chordal Annihilation Sequences. We consider the chordal case first, since all of the remaining classes are subclasses of the chordal graphs. The proofs of the remaining cases are in most cases variations on the proof of the following theorem. As always, it is ultimately the labeling of the vertices that determines the sequence returned. The following theorem was implicit in the work of Rose, Tarjan, and Lueker [14].

THEOREM 1. *Let $G = (V, E)$ be a chordal graph of order p and size q, and let $G_0, G_1, \ldots, G_q$ be the sequence of graphs generated by Algorithm B. If the labeling of the elements of V corresponds to a perfect elimination ordering, then each of $G_1, \ldots, G_q$ is chordal.*

Proof: We show that if α is a perfect elimination ordering for G, then α is a perfect elimination ordering for G_i for each $1 \leq i \leq q$. Assume that $G = G_0$ is chordal, with vertex labels $v_i, v_2, \ldots, v_p$ constituting a perfect elimination ordering α for G. Let $G_1, \ldots, G_q$ be the sequence of graphs generated by the algorithm. If all are chordal, then we are done. Assume by way of contradiction that this is not the case, and let G_k be the first in the sequence for which α is not a perfect elimination ordering. Since G is chordal, $k \geq 1$. Let i be the least index for which v_i is not simplicial in $G_k - \{v_j|j < i\}$.

Algorithm B (Graph Annihilation Favoring Least-Indexed Neighbor)
Input: Graph $G = (V, E)$ of order n, size q, with vertex labeling $\alpha = v_1, \ldots, v_n$.
Output: I_n.

BEGIN
    $G_0 := G$;
    $E_0 := E$;
    $s := |E|$;
    FOR $i := 1$ TO s DO
        BEGIN
            $k_i := \min\{k| \deg_{i-1} v_k > 0\}$;
            $j_i := \min\{j|v_j v_{k_i} \in E_{i-1}\}$;
            $e_i := v_{k_i} v_{j_i}$;
            $E_i := E_{i-1} - e_i$;
            $G_i := (V, E_i)$;
        ENDFOR
END.

FIG. 2. *Annihilation algorithm favoring least-indexed neighbor*

Algorithm C (Annihilation Algorithm Favoring Greatest-Indexed Neighbor)
Input: Graph $G = (V, E)$ of order n, size q, with vertex labeling $\alpha = v_1, \ldots, v_n$.
Output: I_n.

BEGIN
    $G_0 := G$;
    $E_0 := E$;
    $s := |E|$;
    FOR $i := 1$ TO s DO
        BEGIN
            $k_i := \min\{k| \deg_{i-1} v_k > 0\}$;
            $j_i := \min\{j|v_j v_{k_i} \in E_{i-1}\}$;
            $e_i := v_{k_i} v_{j_i}$;
            $E_i := E_{i-1} - e_i$;
            $G_i := (V, E_i)$;
        ENDFOR
END.

FIG. 3. *Annihilation algorithm favoring greatest-indexed neighbor*

Consider $N_k[v_i]$. Since v_i was simplicial in $G_{k-1} - \{v_j | j < i\}$, then there must be some pair of vertices $v_s, v_t \in N_k(v_i)$, with $i < s, t$ and with the property that $v_s v_t \in E_{k-1}$ but $v_s v_t \notin E_k$. This implies that $v_s v_t$ was deleted by the algorithm. But this would imply that $\min\{s, t\} < i$, a contradiction. It follows that α is a perfect elimination ordering for each graph in the sequence, and the proof is complete. □

2.2. Strongly Chordal Annihilation Sequences. While the strongly chordal graphs have a forbidden subgraph characterization, as described earlier, such a characterization is not useful from the algorithmic point of view. More useful is the characterization in terms of strong elimination orderings; the recognition algorithm given by Farber [6] constructs such an ordering in $O(n^2)$ time if such an ordering exists. The input graph must be labeled according to such an ordering.

THEOREM 2. *Let $G = (V, E)$ be a strongly chordal graph of order p and size q, and let $G_0, G_1, \ldots, G_q$ be the sequence of graphs generated by Algorithm B. If the labeling of the elements of V corresponds to a strong elimination ordering, then each of $G_1, \ldots, G_q$ is strongly chordal.*

Proof: We show that if α is a strong elimination ordering for G, then α is a strong elimination ordering for G_i for each $1 \leq i \leq q$. Assume that $G = G_0$ is strongly chordal, with vertex labels $v_i, v_2, \ldots, v_p$ constituting a strong elimination ordering α for G. Let $G_1, \ldots, G_q$ be the sequence of graphs generated by the algorithm. If all are strongly chordal, then we are done. Assume by way of contradiction that this is not the case, and let G_k be the first in the sequence for which α is not a strong elimination ordering. Since G is strongly chordal, $k \geq 1$. Let i be the least index for which v_i is not simple in $G_k - \{v_j | j < i\}$. Consider $N_k[v_i]$. Since v_i was simple in $G_{k-1} - \{v_j | j < i\}$, then there must be some pair of vertices $v_s, v_t \in N_k(v_i)$, with $i < s, t$ and with the property that v_s and v_t are not compatible in G_k but were compatible in G_{k-1}. By Theorem 1 we know that $v_s v_t \in E_k$. Since v_s and v_t were compatible in G_{k-1}, it follows that the algorithm has removed an edge incident to one of these. Since this has occurred before v_i has become isolated, then v_i must be one endpoint of the missing edge. But deleting a vertex from $N(vi)$ can have no effect on the pairwise compatibility of the remaining neighbors of v_i, so we have a contradiction and the result follows. □

2.3. Split Annihilation Sequences. Split graphs have a simple characterization in terms of their structure, but this does not lend itself to direct application in devising algorithms for working with these graphs. Fortunately, other characterizations are available; it turns out that simply ranking the vertices by degree provides the key to a fast recognition algorithm, and it it this degree sequence ordering that we use as input to the algorithm. But in proving the correctness of the result, we fall back on the structural characterization.

THEOREM 3. *Let $G = (V, E)$ be a split graph of order p and size q, with degree sequence ordering α. Let β be the reversal of α. Let $G_0, G_1, \ldots, G_q$ be the sequence of graphs generated by Algorithm B using β as the vertex ordering. Then each of $G_1, \ldots, G_q$ is a split graph.*

Proof: By definition of split graph, the vertices of G can be partitioned as $V = K \bigcup I$, where $\langle K \rangle$ is a maximal clique and I is an independent set. If $x \in I$ and $y \in K$, then clearly $\deg(x) \leq \deg(y)$, with equality possible only if y has no neighbor in I and $N(x) = K - \{y\}$. Partition $I = I_< \bigcup I_=$, where $I_< = \{x \in I | \deg(x) < |K| - 1\}$ and $I_= = \{x \in I | \deg(x) < |K| - 1\}$. If $I_= = \emptyset$, then it is obvious that each graph in the sequence $G_1, \dots, G_p$ is a split graph. Suppose $I_=$ is not empty. Certainly the first $|I_<|$ vertices can be isolated without destroying the split property, since the clique K is preserved. Since $I_=$ is nonempty, then every vertex $x \in I_=$ is adjacent to exactly $|K| - 1$ vertices of K and consequently nonadjacent to exactly one vertex $y \in K$. Since $\deg(x) = \deg(y)$, it is possible that y will be processed before x. But $K - \{y\} + \{x\}$ is a clique, and $I - \{x\} + \{y\}$ is an independent set, so the split property is maintained. It is easy to see that the split property is maintained once the vertices in $I_=$ have become isolated. The result follows. □

2.4. Threshold Annihilation Sequences. Threshold graphs are a special case of the split graphs. Like the class of split graphs, the class of threshold graphs is closed under the taking of complements. For applications and characterizations, see Golumbic [8]. The input ordering is by vertex degree, as in the case of split graphs. The characterization used in the proof of the following theorem is simple: every connected threshold graph has at least one dominating vertex, and since the threshold property is hereditary then it follows that the subgraph obtained by deleting a dominating vertex is threshold and therefore has a dominating vertex of its own.

THEOREM 4. *Let $G = (V, E)$ be a threshold graph of order p and size q, and let $G_0, G_1, \dots, G_q$ be the sequence of graphs generated by Algorithm B. If the labeling of the elements of V corresponds to a threshold elimination ordering, then each of $G_1, \dots, G_q$ is a threshold graph.*

Proof: We show that if α is a threshold elimination ordering for G then α is a threshold elimination ordering for G_i, $1 \leq i \leq p$. Let k be the least index such that G_k is not threshold. Let r be the greatest index such that v_r is not a dominating vertex in $G_k - \{v_i | i > r\}\}$. By our hypothesis, there exists some vertex $s < r$ such that $v_r v_s$ was an edge in G_{k-1} but was deleted at the most recent iteration of the algorithm. Moreover, it must be the case that v_s has positive degree in G_k, since we are not concerned with isolated vertices. But if $v_r v_s$ has just been deleted, it must be the case that $r = \min\{t | v_s v_t \in E_{k-1}\}$, in which case v_r is in fact a dominating vertex in $G_k - \{v_i | i > r\}$ and the proof is complete. □

2.5. Interval Annihilation Sequences. Interval graphs have a number of characterizations, some of which are described earlier in this report. The characterization that we use here is that every interval graph possesses (and no noninterval graph possesses) an *interval elimination ordering*, which is an ordering of the vertices as $v_1, v_n, \dots, v_p$ with the property that if $i < j < k$ and if $v_i v_k \in E$ then $v_j v_k \in E$. This characterization is, in a sense, a restatement of the characterization due to Fulkerson and Gross [7].

THEOREM 5. *Let $G = (V, E)$ be an interval graph of order p and size q, and let $G_0, G_1, \ldots, G_q$ be the sequence of graphs generated by Algorithm B. If the labeling of the elements of V corresponds to an interval elimination ordering, then each of $G_1, \ldots, G_q$ is an interval graph.*

Proof: Let $G = (V, E)$ be as described. Assume that the labeling of the vertices corresponds to an interval elimination ordering. Let $m = \min\{j | \alpha \text{ fails for } G_j\}$. By definition of interval elimination ordering, it follows that there exist vertices v_i, v_j, v_k, with $i < j < k$, such that $v_i v_k \in E_m$ but $v_j v_k \notin E_m$. Since α was an interval elimination ordering for G_{m-1}, then $v_j v_k \in E_{m-1}$. But then the algorithm deleted $v_j v_k$, which implies that $j < i$, and we have a contradiction. □

2.6. Unit Interval Annihilation Sequences. This is the only case requiring Algorithm C. In all other respects, the proof is similar to its predecessors. The ordering that we use is a *bicompatible elimination ordering*, which is most succinctly described as an labeling of the vertices with the remarkable property that both $v_1, v_2, \ldots, v_p$ and $v_p, v_{p-1}, ldots, v_1$ are perfect elimination orderings.

THEOREM 6. *Let $G = (V, E)$ be a unit interval graph of order p and size q, and let $G_0, G_1, \ldots, G_q$ be the sequence of graphs generated by Algorithm C. If the labeling of the elements of V corresponds to a bicompatible elimination ordering, then each of $G_1, \ldots, G_q$ is a unit interval graph.*

Proof: Suppose $G = (V, E)$ is a unit interval graph as described, with bicompatible elimination ordering α. Denote by β the reversal of α. By definition of bicompatible ordering, both α and β are perfect elimination orderings. Let G_m be the first in the sequence of graphs produced by the algorithm for which α is not a bicompatible ordering. Let $v_r v_t$ be the edge most recently removed by the algorithm, and suppose $r < t$. We have two possibilities. First suppose that α is not a perfect elimination ordering for G_m. By hypothesis, α was a perfect elimination ordering for G_{m-1}. Since $v_r v_t$ has just been deleted, we know that the set $\{v_i | i < r\}$ is an independent set in G_m, so the failure of α for G_m implies that there is some v_s, with $r \leq s$, such that v_s is not simplicial in $G_m - \{v_i | i < s\}$. Since $v_r v_t$ is the edge that caused the trouble, it follows that $r \geq s$, from which we have $r = s$. But deleting an edge incident to v_r cannot affect the simpliciality of v_r, so α turns out to be a perfect elimination ordering for G_m after all. The problem must lie with β. For the deletion of $v_r v_t$ to cause β to fail as a perfect elimination ordering for G_m it must be that v_w, where $w > t$, is not simplicial in $G_m - \{v_i | i > w\}$. Since this is not the case in G_{m-1}, it follows that $v_r \in N_m(v_w)$. But this is impossible, since the algorithm would have deleted $v_r v_w$ before deleting $v_r v_t$. So β is a perfect elimination ordering for each graph in the generated sequence, and the result follows. □

3. Complexity of Constructing Annihilation Sequences. Since algorithms B and C are both quadratic in the number of vertices (linear in the number of edges), we know that the cost of constructing annihilation sequences is $\Omega(n^2)$ for graphs of order n, provided the input graphs are classified *a priori* as chordal, etc., and that the

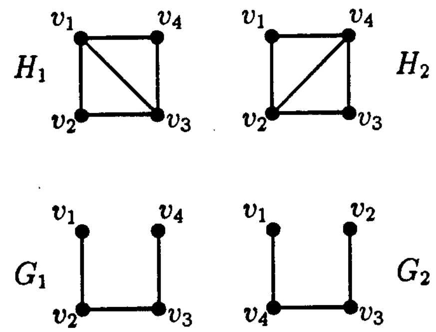

FIG. 4. *The least upper bound of G_1 and G_2 is undefined.*

associated elimination orderings are in hand. In the chordal, strongly chordal, split, and threshold cases, we can construct these orderings in $O(n^2)$ time, while in the interval and unit interval cases we must work harder, obtaining the desired orderings in $O(n^3)$ time.

4. A Partial Ordering of the Chordal Graphs. If we let $G^{(n)}$ denote the set containing all graphs of order n, and if we say for any graphs $G, H \in G^{(n)}$ that $G \preceq H$ if $E(G) \subseteq E(H)$, then it is easy to see that $(G^{(n)}, \preceq)$ is a partially ordered set. In fact, $(G^{(n)}, \preceq)$ is a lattice. There is nothing particularly astonishing in this observation; if we denote by $V^{(2)}$ the collection of all order-2 subsets of V, then every graph with vertex set V has for its edge set some subset of $V^{(2)}$. It follows that $(G^{(n)}, \preceq) \cong (\mathcal{P}(V^{(2)}), \subseteq)$. It is clear from the work of Grone, Johnson, et al. [9] that every chordal graph allows at least one completion sequence, and implicit in their work that every chordal graph allows at least one annihilation sequence. These phenomena combined reveal that the chordal graphs of order n constitute a subposet of $(G^{(n)}, \preceq)$. This subposet is not a sublattice, since if G and H are chordal graphs of order n then $\inf(G, H)$ and $\sup(G, H)$ might not be chordal. If we view the chordal subposet as a poset in its own right and ignore the embedding in $(G^{(n)}, \preceq)$, then it is possible for graphs G and H to have competing upper (lower) bounds. For a simple example, consider the two elements $G_1, G_2 \in G^{(4)}$ as shown in Figure 4. Each is isomorphic to P_4, a path on four vertices. In $(G^{(4)}, \preceq)$, their supremum is found by taking the union of their respective edge sets. The result, a four-cycle, is not chordal. In fact, G_1 and G_2 have no common chordal supergraph of size four. Their nearest common chordal supergraphs are of size five, and are labeled H_1, H_2 in the same figure. Neither contains the other, so $\sup(G_1, G_2)$ is undefined. Nevertheless, it is certain (by construction) that every chordal graph lies on at least one maximum chain in the chordal poset.

So the chordal graphs of order n are broadly distributed within the less organized set of general graphs of order n. For any $0 \leq k \leq \binom{n}{2}$, there is at least one chordal graph of order n with k edges. For n close to the middle of this range, there are many such graphs. Since the strongly chordal graphs are themselves chordal, we see that the strongly chordal graphs constitute a subposet of the chordal graphs, as do the interval graphs. The unit interval graphs are of course interval, so they constitute a subposet of the interval graphs, as do the threshold graphs. The split graphs constitute another

subposet of the chordal graphs, and the threshold graphs turn up as a subposet of the split graphs, as well. This a surprisingly rich structure that we find embedded in the relatively uninteresting lattice $(G^{(n)}, \preceq)$, and it is reasonable to anticipate that a deeper understanding of this structure will be of use in designing improved algorithms for finding approximate solutions to real-world problems.

5. Directions for Further Work. The strongly chordal graphs are associated with a particular family of integral polyhedra, and with a related family of 0, 1-matrices called *totally balanced*. It will be interesting to discover what, if any, significance the strong completion and annihilation sequences have when viewed through a "polyhedral lens."

We have confirmed a conjecture, first stated in [10], that the classes under consideration are each partially ordered under edge set inclusion. Particularly in the chordal case, this phenomenon might be of use in developing heuristics to solve general completion problems in which one seeks a graph with some property P that is as close as possible in some metric to a particular input graph G. Problems that must be resolved are numerous. For example, we need a measure of distance between an arbitrary graph G and the chordal poset. What could be meant by "nearest chordal neighbor"? Intuitively, it is tempting to use for this measure the minimum cardinality of the symmetric differences of the edge set of the graph in question and the edge sets of each chordal graph at the same level in the poset; this is essentially taking the Hamming distance between characteristic vectors of edge sets. Minimal chordal supergraphs and maximal chordal subgraphs have enjoyed some consideration in heuristics for solving coloring, scheduling, and related problems ([4]), and we are optimistic that these new results can be brought to bear on the same problems. Another obvious question is one of navigation, in the following sense: suppose G is not chordal, and we are faced with finding a near-optimal coloring of G. Suppose furthermore that we have in hand a chordal graph H that is, say, a maximal chordal subgraph of G. We would like to exploit the availability of cheaply-generated completion sequences to traverse a chordal chain upward until we reach a chordal supergraph of G, but *which chain do we choose*? More precisely, in view of the algorithm's dependence on the vertex labels, which vertex labeling do we choose? We suspect that this is itself an NP-complete problem but that heuristics might be devised to find productive, if suboptimal, choices. For example, we might consider using a genetic algorithm to choose promising labelings of the vertex set, or adopt a strategy that assigns vertex labels in ascending or descending order by vertex degree.

In addition to pursuing applications of these results, the substructures embedded in $(G^{(n)}, \preceq)$ and revealed here merit further study.

REFERENCES

[1] J. A. BONDY AND U. S. R. MURTY, *Graph Theory with Applications*, North-Holland, 1976.

[2] T. CARROLL, *Edge annihilation sequences for classes of chordal graphs*, Master's thesis, Naval Postgraduate School, June 1996.

[3] V. CHVÁTAL AND P. HAMMER, *Set-packing and threshold graphs*, Research Report CORR 73-21, University of Waterloo, 1973.

[4] P. M. DEARING, D. R. SHIER, AND D. D. WARNER, *Maximal chordal subgraphs*, Discrete Applied Mathematics, 20 (1988), pp. 181–190.

[5] G. A. DIRAC, *On rigid circuit graphs*, Abh. Math. Sem. Univ. Hamburg, 25 (1961), pp. 71–76.

[6] M. FARBER, *Characterizations of strongly chordal graphs*, Discrete Mathematics, 43 (1983), pp. 173–189.

[7] D. R. FULKERSON AND O. A. GROSS, *Incidence matrices and interval graphs*, Pacific Journal of Mathematics, 15 (1965), pp. 835–855.

[8] M. C. GOLUMBIC, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, 1980.

[9] R. GRONE, C. R. JOHNSON, E. M. SÁ, AND H. WOLKOWICZ, *Positive definite completions of partial hermitian matrices*, Linear Algebra and Its Applications, 58 (1984), pp. 109–124.

[10] R. M. ODOM AND C. W. RASMUSSEN, *Conditional completion algorithms for classes of chordal graphs*, Congressus Numerantium, 103 (1994), pp. 183–192.

[11] C. W. RASMUSSEN, *Conditional graph completions*, Congressus Numerantium, 103 (1994), pp. 183–192.

[12] F. S. ROBERTS, *Indifference graphs*, in Proof Techniques in Graph Theory, F. Harary, ed., Academic Press, New York, 1969, pp. 139–146.

[13] ——, *Discrete Mathematical Models*, Prentice-Hall, 1976.

[14] D. ROSE, R. TARJAN, AND G. LEUKER, *Algorithmic aspects of vertex elimination on graphs*, SIAM Journal of Computing, 5 (1976), pp. 266–283.

[15] D. B. WEST, *Introduction to Graph Theory*, Prentice Hall, 1996.

DISTRIBUTION LIST

Director Defense Technical Information Center 8725 John J. Kingman Rd., STE 0944 Ft. Belvoir, VA 22060-6218	(2)
Research Office Code 09 Naval Postgraduate School Monterey, CA 93943	(1)
Library Code 52 Naval Postgraduate School Monterey, CA 93943	(2)
Professor W.M. Woods, Code MA Department of Mathematics Naval Postgraduate School Monterey, CA 93943	(1)
Craig W. Rasmussen, Code MA/Ra Department of Mathematics Naval Postgraduate School Monterey, CA 93943	(3)
LCDR Thomas Carroll, USN VAW 120 1027 Bellinger Rd. NAS Norfolk, VA 23511	(3)
Dr. Marc Lipman Mathematical, Computer and Information Sciences Division Office of Naval Research 800 Quincy St. Arlington, VA 22217-5000	(1)